

# ***The fast tracker processor for hadron collider triggers, IEEE Transactions Nuclear Science***

**Alberto Annovi**

Istituto Nazionale di Fisica Nucleare

**Maria Grazia Bagliesi**

Istituto Nazionale di Fisica Nucleare

**Antonio Bardi**

Dipartimento di Fisica

Università di Pisa

**Roberto Carosi**

Istituto Nazionale di Fisica Nucleare

**Mauro Dell'Orso**

Dipartimento di Fisica

Università di Pisa

A. Annovi, M.G. Bagliesi, A. Bardi, R. Carosi, M. Dell'Orso, M. D'Onofrio, P. Giannetti, G. Iannaccone, F. Morsani, M. Pietri, G. Varotto, *The fast tracker processor for hadron collider triggers*, IEEE Transactions Nuclear Science, **48**(3), p. 575 (2001).

**M. D'Onofrio**

Dipartimento di Fisica  
Università di Pisa

**Paola Giannetti**

Istituto Nazionale di Fisica Nucleare

**Giuseppe Iannaccone**

Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni,  
Università di Pisa

**Fabio Morsani**

Istituto Nazionale di Fisica Nucleare

**Marco Pietri**

Istituto Nazionale di Fisica Nucleare

**Graziano Varotto**

Istituto Nazionale di Fisica Nucleare

# The Fast Tracker Processor for Hadronic Collider Triggers

A. Annovi<sup>2</sup>, M.G. Bagliesi<sup>2</sup>, A. Bardi<sup>2</sup>, R. Carosi<sup>2</sup>, M. Dell'Orso<sup>1,2</sup>, M. D'Onofrio<sup>1,2</sup>, P. Giannetti<sup>2</sup>, G. Iannaccone<sup>3</sup>, F. Morsani<sup>2</sup>, M. Pietri<sup>2</sup>, G. Varotto<sup>2</sup>

<sup>1</sup>Dipartimento di Fisica, Università di Pisa, Piazza Torricelli 2, 56100 Pisa, Italy

<sup>2</sup>INFN Pisa, Via Livornese 1291, 56010 S. Piero A Grado (PI), Italy

<sup>3</sup>Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Via Diotisalvi 2, 56126 Pisa, Italy

## Abstract

Perspective for precise and fast track reconstruction in future hadronic collider experiments are addressed.

We discuss the feasibility of a pipelined highly parallelized processor dedicated to the implementation of a very fast algorithm. The algorithm is based on the use of a large bank of pre-stored combinations of trajectory points (patterns) for extremely complex tracking systems. The CMS experiment at LHC is used as a benchmark. Tracking data from the events selected by the level-1 trigger are sorted and filtered by the Fast Tracker processor at a rate of 100 kHz. This data organization allows the level-2 trigger logic to reconstruct full resolution tracks with transverse momentum above few GeV and search secondary vertexes within typical level-2 times.

## I. INTRODUCTION

In this paper we describe the implementation of the Fast Tracker (FTK) [1], a highly parallelized processor dedicated to the efficient execution of a fast track finding algorithm [2], based on the idea of a large bank of pre-calculated hit patterns [3]. We estimate the size of the hardware necessary to apply this technique in very complex tracking systems: the CMS experiment at LHC [4] is used as a benchmark.

The proposed system is an evolution of the Silicon Vertex Tracker (SVT) [5] currently being built for the CDF experiment. The CDF tracker processes data with a 100 kHz input rate, and an overall allowed processing time (latency) of 10  $\mu$ s. Five layers from the silicon vertex detector can be linked to segments observed in the drift chamber to reconstruct real time tracks precise enough to measure, for instance,  $b$  quark decay vertexes.

The long latency time available at the future experiments allows extensive pipelining in order to subdivide the complex pattern recognition into simpler sequential steps with increasing resolution. The pattern recognition consists in associating hits into track candidates at low resolution (*roads*), then the tracks are fitted and their parameters precisely determined. If the hardware is enough powerful this work can be divided into only two steps: the first step, the *road finding*, is executed by the FTK processor, the second step, the *track fitting*, can be executed by any kind of level-2 trigger logic fast enough to work in pipeline with FTK.

Figure 1 shows FTK spying the whole amount of tracking data at a very high rate (up to 100 kHz) to perform data reduction for trigger applications. FTK performs the most CPU consuming part of the pattern recognition. It subdivides the enormous problem of finding tracks inside the entire detector

into many simpler problems of track finding inside roads with width of  $10^{-2}$ –10 cm. Hits of roads, with transverse momentum ( $P_T$ ) above a threshold of few GeV/c can be filtered among a huge number of other hits and organized in memories available to the level-2 trigger logic, where the pattern recognition should be completed. The level-2 logic should find real tracks inside roads and calculate track-based physical quantities, such as invariant masses and decay vertexes. The road width must be optimized for the characteristics of the specific experiment. Too small or too great widths would require intolerably high performances respectively to FTK or to the level-2 logic.

## II. FTK PROCESSOR ARCHITECTURE

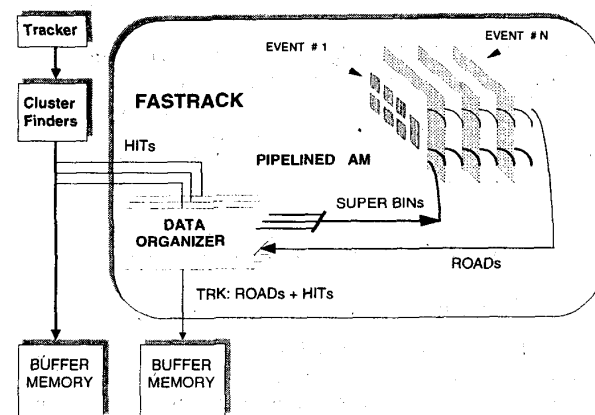


Figure 1: The FTK processor spies the tracking data of level-1 selected events to produce an organized memory where only candidate tracks with  $P_T$  above few GeVs are written.

The FTK processor is composed of two cooperating parts: the Data Organizer (DO) and the pipelined Associative Memory (AM).

The Data Organizer [6] is the interface with the DAQ. It performs the following tasks: (a) It receives full resolution detector hits and buffers them in an internal database. (b) It sends low-resolution hits called Super Bins (SB) to the AM pipeline. The Super Bins are obtained by logically ORing a number of adjacent detector bins. (c) It receives back roads from the AM and fetches from the internal data base all the detector hits contained in the roads. (d) It sends each road with its set of full resolution hits to a memory accessed by the level-2 trigger logic. The Data Organizer can perform this non-

trivial task at full rate, in the same time needed for a simple buffering function.

The pipelined Associative Memory implements the algorithm that finds roads. The AM is a dedicated device where parallelism is pushed to the maximum level since each stored hit pattern is provided with the necessary hardware to compare itself with the event. The AM pattern bank is limited by the size of the hardware, mainly consisting of low-density custom memories. Pattern banks for the tracking detectors of the next generation hadron colliders are very large. As an example, we estimate the bank size for the barrel of the CMS experiment (see section IV.B). We suppose to use four independent AMs, each one working on a fourth of the barrel. This segmentation generates some inefficiencies at the sector boundaries (see section IV.D), but it is necessary because the barrel amount of tracking data is too large to be brought to a single AM with an event rate of 100 kHz. In fact the detector area searched by a bank is limited by the AM bandwidth. For this aspect, FTK has a bandwidth a factor 10 larger than SVT at CDF. All the CDF data sent to the AM are serialized on a single data bus, 15 bits wide, where all the hits flow at a rate of 30 MHz for a total of 0.45 Gbit/s. This serialization requires a segmentation of the central detector into twelve sectors. FTK increases the data flow rate exploiting a parallel readout of the detector layers. Data are fed in the new AM on six parallel buses with a rate of 100 bits every 25 nsec for a total of 4 Gbit/s.

In order to warrant scalability of the architecture, the whole AM bank is a pipeline of AM boards: Hit data feed all AM boards one after the other. In such a way boards can be simply added, with the only drawback of increased data latency. The AM pipelined structure allows to expand the sector bank size as necessary without any timing degradation.

### III. HARDWARE IMPLEMENTATION

#### A. The FTK brain: AM-board and DO-board

Both the Data Organizer and the Associative Memory are composed of a set of 9U VME boards, the DO-boards and the AM-boards. Each DO-board receives hits from one or two detector layers and sends them with the proper resolution on a single bus to the pipelined AM. The choice between one or two detector layers is made on the basis of the mean layer occupancy. The AM-board can receive up to six independent buses from six DO-boards, to perform pattern recognition with up to twelve layers.

Both DO and AM boards are implemented using modern and powerful programming devices (CPLD and FPGA devices from Xilinx [7]). Each board is synchronized by an internal clock signal and works up to 40 MHz. Each board input is provided of a synchronous FIFO where read & write functions are asynchronous and totally independent. These FIFOs allow communication between asynchronous boards: the write function is synchronous with the upstream board and the read function with the board that has the FIFO.

Both AM-board and DO-board represent a significant technological challenge. They are very complex projects and their difficulty concerns very different technological aspects:

1. AM-board has a very regular structure characterized by a small amount of logic (the basic element of the associative memory) repeated many times. The technological challenge is due to the necessity to pack as much patterns as possible on a board and to distribute all detector hits to all the patterns. Six hit buses must reach each AM chip on the board and each pattern on the chip. For this reason a significant effort has been dedicated to the chip density study (see next section) and to a very dense board construction. A 9U VME board allocates 128 AM chips, whose pattern capacity strongly depends on technological choices (see next section). An AM-board consists of 4 identical smaller boards (LAMB board) operating in parallel, each containing a pipeline of 32 AM chips. For details on board construction see [8].
2. DO-board is characterized by a large amount of very complex logic: data flow in a long pipeline under the control of many cooperating finite state machines and a lot of auxiliary logic that manipulates data on the fly. The attempt to place the whole amount of logic in a single very large FPGA failed, since we couldn't reach the wanted speed. We needed to divide and optimize the logic into many different programmable chips. Each single chip needed at least an independent optimization effort, sometimes more than one, since the optimization of a project piece, often causes changes also in other parts. We used low-density, very fast CPLD devices (XC9500XV) for very complex equations (finite state machines as example) and high-density FPGAs (Virtex) for logic too rich of registers (the data pipeline as example). For details on board construction see [6].

#### B. Packing Patterns inside a 9U VME board

In order to estimate the hardware complexity of the AM bank (the necessary number of boards) we have considered two possible implementations: one based on a commercial low-cost FPGA family (Xilinx Spartan 0.35  $\mu$ m process) [8] and the other based on standard cell ASICs [9]. The FPGA approach allows a larger degree of flexibility in the prototyping and testing phase of the project. It also allows easy upgrade of the system when new generations of FPGA will be delivered to the market: indeed, newer pin-compatible FPGAs can replace old chips in the same PCB, and be very conveniently configured via the high-level hardware description of the logic. On the other hand, the ASIC approach is optimized in terms of delays and pattern density.

For both approaches, we consider a fully modular architecture in which each module stores a single trajectory and contains the logic needed to compare the coordinates of all fired detectors with those associated to the stored trajectory. All modules in each AM chip (FPGA or ASICs) receive from six input buses (one bus for a pair of layers) the complete configuration of fired detectors of each event. We consider that a single trajectory consists of a 18-bit word for each of the 12 layers. The most significant bit identifies the layer on each bus.

Using the 0.35  $\mu$ m FPGA family mentioned above, 56 patterns can be allocated on a chip, and 128 chips with the

PQ208 package can be allocated on a 9U VME board, for a total number of 7168 patterns per board. This is obtained with a very careful mapping of the logical functions onto the FPGA, with 95% of the available Configurable Logic Blocks of each chip actually used. For details on the chip design and layout see reference [8].

Estimates for the year 2005 (the scheduled starting date for next generation hadronic collider experiments) can be made on the basis of data from the FPGA manufacturer and from the 1998 edition of the International Technology Roadmap for Semiconductors [10].

According to Xilinx, in 2005 the low-end family will be based on a 0.13  $\mu\text{m}$  process. Assuming that CLB area scales with technology, and that again 95% of the CLBs will be used, a density will be reached of  $6.4 \cdot 10^4$  patterns per board. On the other end, the high-end FPGA family (Virtex), which has a logical structure very similar to Spartan, will be based on the more advanced 100 nm process, and will allow a density of  $3.3 \cdot 10^5$  patterns per board.

As far as the ASIC implementation is concerned, according to the ITRS [10], in 2005 the 0.1  $\mu\text{m}$  process will be available for ASICs. Based on the architecture proposed in [10], re-scaled for considering six 18-bit input buses, 16 mm<sup>2</sup> will be required for a single pattern. This means that using the PQ208 package (so that 128 chips can be allocated on a single AM board), we will be able to reach a density of  $5 \cdot 10^6$  patterns per board.

The FPGA approach, though rather limited in terms of performance and pattern density, will be very useful for prototyping and for testing the complete architecture of the system. Close to the starting date of the actual experiment, one can switch to an ASIC implementation based on the most recent technology process, for increasing the packing density of more than one order of magnitude and therefore reducing the hardware to a reasonably manageable size.

### C. The crate layout

Figure 2 shows how FTK is organized inside a VME crate. The two important sections are the set of DO-boards and AM-boards.

Up to six DO-boards are expected to transmit hit data to the AM pipeline on six independent buses. However the number of DO-boards in the system can be larger. Each layer (or pair of layers if they can be handled together) whose data needs to be used in the *track fitting* step needs a DO-board even if the same layer is not used by the AM for the *road finding*. In this case the DO-board won't send SBs to the AM pipeline, but will receive roads back from it, and will save the relative full resolution hits in the memory available to the level-2 trigger logic. Up to twelve DO-boards can be allocated in the system.

The back of the DO-board is dedicated to the connection with the DAQ [6]. A flat cable attached on the front panel is used both for the Super Bin bus to the AM pipeline and the TRK bus to the external memory (see figure 1). The six SB buses are received in the front panel by a very simple board (DO-AM interface). It transfers them on the back of the crate where a dedicated back-plane allows a clean propagation of the six buses in the AM pipeline [8]. The TRK buses that are as many as many DO-boards are in the system (up to twelve)

are collected by the Ghost Buster board that eliminates duplicated roads and merge the many DO-board outputs in a single stream. Duplication of roads can happen if candidate tracks are accepted also with missing hits (allowing for missing points reduces the effect of detector inefficiencies). In this case two found roads that differ only for the missing points, will appear identical for the track fitting step.

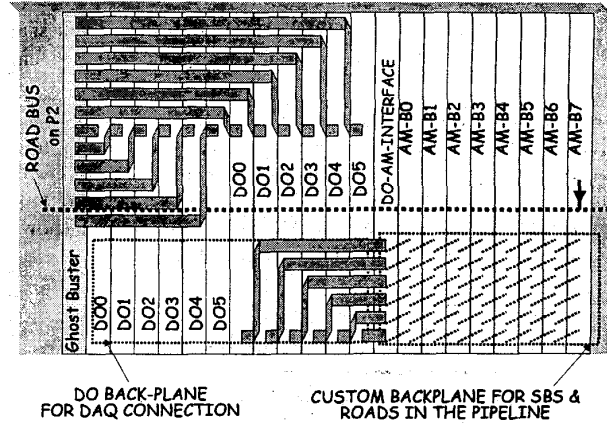


Figure 2: Sketch of the FTK VME crate.

Finally, Roads from the last AM-board in the pipeline are broadcasted to all DO-boards using the USER dedicated bus lines of the VME P2 connector [8].

## IV. TRACK FINDING PERFORMANCES

### A. FTK and experiment simulation

The performance of the algorithm has been studied using the CMS central detector [11] ("barrel") as a benchmark. Taking into account the symmetry of the detector, only a  $\Delta\phi$  sector of  $90^\circ$  for  $z > 0$  is considered. All results reported in the following sections are relative to 1/8 of the whole barrel.

A standalone simulation program has been used to generate tracks in the CMS detector. It takes into account effects such as multiple scattering, ionization energy loss, non-uniformity of the magnetic field, detector inefficiencies and resolution smearing. Detector hits are produced from tracks generated in two different ways:

- Low-LUM sample: Standard Model Higgs events (HTT processes) were simulated with Pythia version 6.125 [12], for an Higgs mass of 120 GeV/c. Random hits were added to the event to take into account the detector noise and two Minimum Bias events. This is the average number of events that overlaps the hard scattering in the LHC low luminosity run. HTT events have an average number of 120 tracks with  $P_T$  above 2 GeV in the barrel, distributed in very energetic jets. It is an example of very crowded events, where pattern recognition could be particularly difficult.
- High-LUM sample: 360 energetic tracks per event  $P_T$  above 5 GeV) were generated in the barrel uniformly

distributed in  $\phi$ . Random hits were added to take into account detector noise and 30 Minimum Bias pile-up events, more than the average number of soft collisions per beam crossing in the LHC high luminosity run. This kind of events is more complex than most of the physical ones and we consider it an upper-limit case. It constitutes a very severe test for the online track finding project.

The event primary vertex is smeared using a gaussian distribution of  $\sigma = 1\text{mm}$  in the transverse plane and  $\sigma = 3\text{cm}$  along  $z$ .

Only 8 cylindrical layers are used to find tracks. Two different choices are tested. The first set is composed by four Silicon layers linked to the most internal four MSGC layers and the second set includes two pixel layers linked to four Silicon layers and two internal MSGC layers. Table 1 shows the distance from the beam line for all used layers.

Table 1  
Distance of used layers from the beam line.

Pixel 1	7 cm
Pixel 2	11 cm
Silicon 1	23.2 cm
Silicon 2	30.9 cm
Silicon 3	38.7 cm
Silicon 4	46.1 cm
MSGC 1	64.1 cm
MSGC 2	72 cm
MSGC 3	79.9 cm
MSGC 4	87.9 cm

The second choice is preferred since composed of more internal layers, more efficient in hadrons detection [11]. The two different layer selections gave very similar results, so in the following we report only results for the pixel configuration.

We require 6 out of 8 layers to be fired to include the hit combination in the candidate track list. We allow for two missing points, to reduce the effect of detector inefficiencies.

The simulation program performs two subsequent steps to reconstruct the event, as the hardware should do:

- *road finding*: all roads are found comparing the event with the pre-stored patterns, simulating the FTK processor work;
- *track fitting*: all found roads are processed to find the best track parameter values and to reject the fake ones. A test is performed on every track candidate inside each road to reject the combinatorial background, and track parameters are calculated. This is achieved using linear approximations for the track constraints and Principal Components Analysis [13]. The precision of the method has been checked in many conditions [14]. It results to be comparable with the full offline resolution, with the additional advantage that calculation is very fast.

## B. The pattern bank size

In principle, the pattern bank may contain all the possible tracks that go through the detector (a 100% efficient bank). In practice, since one should also consider effects which make a particle to deviate from the ideal trajectory (detector resolution smearing, multiple scattering, etc.) and that generate also very low probability patterns, the size of such a bank could be almost impossible to handle. For this reason we decide to use a bank that is partially inefficient. We generate tracks in the detector and we store new patterns corresponding to the generated tracks, until the bank reaches the wanted efficiency. This procedure is slow but the computation is done once forever. It automatically ensures that the high-probability patterns are stored and the low-probability patterns are left out. A reference "bank efficiency" has been fixed at 90%.

The generated track typology also affects a lot the bank size. It is particularly convenient to restrict the range of the generated track parameters, such as  $P_T$  and the region where they came from, the *luminosity region*, to those values which are relevant for the physical processes to be studied. We are interested to be very efficient for tracks above a certain  $P_T$  threshold (for sure we want to reject tracks below 2 GeV/c) and coming from the interaction region.

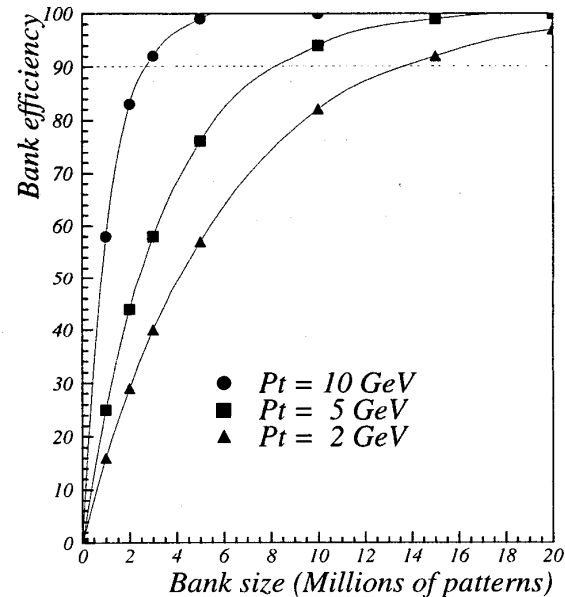


Figure 3: Bank efficiency (%) as a function of the bank size for  $P_T$  thresholds 2 GeV (triangles), 5 GeV (squares) and 10 GeV (circles).

We assume a cylindrical luminosity region, circular in the transverse plane with a radius of 1 mm and 3 cm long in the longitudinal direction. This restriction helps to keep the size of the pattern bank small, but reduces to zero the efficiency for tracks coming from long lived particles. For example K-meson decay products cannot be detected using a pattern bank built with this constraint. However B-meson decay products whose

impact parameters are few hundred  $\mu$ 's are compatible with such luminosity region and pattern bank.

The track  $P_T$  threshold is a very important parameter since it influences the efficiency in collecting interesting events. We would like to keep this threshold as low as possible. Three possible values (2, 5 and 10 GeV/c) have been used to evaluate the size of the corresponding pattern banks. The Super Bin size is another parameter to be studied carefully, since it is critical for the processor performances and for the pattern bank size. It should scale roughly with the detector resolution, therefore in our study we use different values for Silicon (resolution 15  $\mu$ m) and MSGC detectors (resolution 40  $\mu$ m). Three choices have been considered: (1) 1 mm in Si detector and 3 mm in MSGC; (2) 2 mm in Si detector and 5 mm in MSGC; (3) 5 mm in Si detector and 10 mm in MSGC. The segmentation in  $z$  is the same for Si detector and MSGC and it is 8 cells of 12.5 cm for  $z > 0$ . The simulated detector covers a pseudo-rapidity region  $0 < \eta < 1$ .

The size of the pre-calculated pattern bank has been studied for every Super Bin size and  $P_T$  threshold. Figure 3 shows the bank efficiency versus the bank size for various  $P_T$  thresholds (2, 5 and 10 GeV/c) when the Super Bin sizes are 1 mm in the Silicon detectors and 3 mm in the MSGC's.

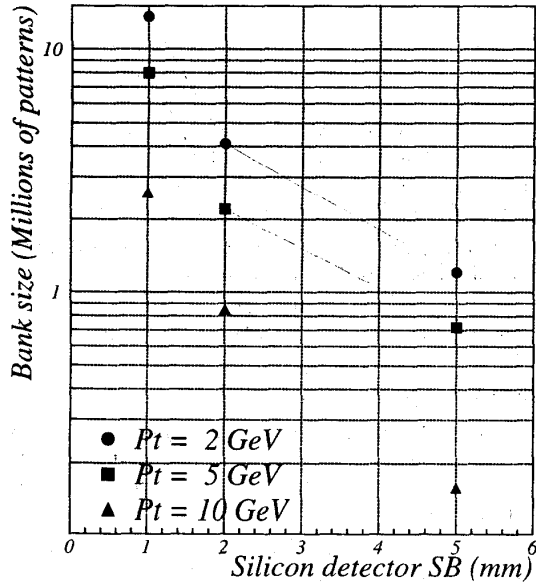


Figure 4: 90% efficient Bank size as a function of the silicon detector Super Bin size (mm) for  $P_T$  thresholds 2 GeV (circles), 5 GeV (squares) and 10 GeV (triangles).

Figure 4 shows the 90% efficient bank size as a function of the Super Bin sizes for various  $P_T$  thresholds. In section III.B we have shown that all these bank sizes are affordable. To decide if it is really necessary to push it to the maximum value, we need to evaluate the amount of work that has to be done by the logic working in pipeline with the FTK processor. We will choose the minimum bank size that is compatible with secondary vertexes finding at a rate above 10 KHz.

### C. Finding tracks at full resolution

Because of the Super Bin size, a road may contain physical hits belonging to different particles. Also, depending on the Super Bin size, there is a level of combinatorial background (fake roads). The number of found roads is always bigger than the expected number of tracks. The excess of roads is high when the Super Bin sizes are large and the  $P_T$  threshold low, and viceversa.

These are the most important quantities to evaluate the remaining amount of work needed to refine track finding inside roads:

- $\langle N_{roads}/track \rangle$ : the average ratio between the found number of roads and the expected number of tracks per event;
- $\langle N_{combinations}/road \rangle$ : the average number of hit combinations per road.

Figure 5 shows them as a function of the Super Bin size. The  $P_T$  threshold is fixed at 2 GeV/c. Results are reported for the High and Low -LUM samples. We observe that the thinnest SB size minimizes the differences observed between the two samples.

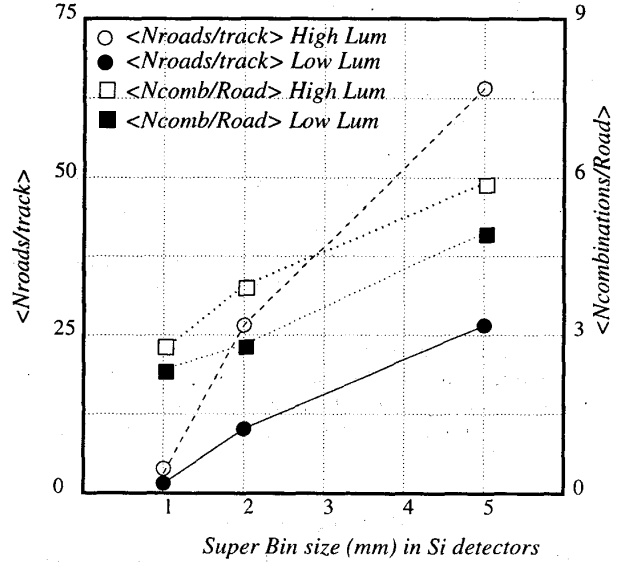


Figure 5: Left scale (full and empty dots): average ratio between the number of found roads and the number of real tracks as a function of the silicon detector Super Bin size (mm); Right scale (full and empty squares): average number of hit combinations per road as a function of the silicon detector Super Bin size (mm). Results are reported for both the High and Low-LUM samples.

The track fitting time is proportional to the quantities  $\langle N_{roads}/track \rangle$  and  $\langle N_{combinations}/road \rangle$ . For example we consider the Low-LUM sample: for Super Bin size 5-10 mm and  $P_T$  threshold 2 GeV, for every input track the track fitter has to check 4.9 combinations times 25 roads per track (125 combinations per track), while for Super Bin size 1-3 and  $P_T$  threshold 2 GeV this number is 2.4 combinations times 1.5 roads = 3.6. We expect therefore the fit to be faster by about a factor 35 in the latter case. In this better case the fitting time

has been checked with an SGI R10000 processor [15], by sequentially fitting the residual combinations of hits in the roads, choosing the best fit and calculating the track parameters. About  $10^5$  tracks per seconds can be reconstructed. This means that a single CPU can reconstruct complex events with 100 tracks of  $P_T$  above 2 GeV (we remind that the very complex HTT events have 120 tracks above this threshold) at a rate of 1000 Hz.

After the track fitting stage the number of found tracks (i.e. the track candidates which pass the  $\chi^2$  cut) is compared to the expected number of track per event in order to evaluate the final number of fake tracks after fit. We find 0.8% fake tracks for the smallest Super Bin sizes (1mm and 3mm) and about 1.8% for the largest Super Bins.

In conclusion we think that the thinnest road is the best choice to have a very fast pattern recognition that can work at high event rates. Figure 5 shows that even the worse conditions (High LUM sample) can be handled with such road sizes. This choice of the road width for a low  $P_T$  threshold of 2 GeV/c, corresponds to a bank size of **3  $10^7$  patterns** per CMS barrel fourth (see figures 3 and 4 where the bank size for 1/8 of barrel is reported). Taking into account the pattern densities evaluated in section III.B for 2005 ASIC technology (5  $10^6$  patterns per board) we can conclude that such bank will occupy 6 slots in a VME crate.

#### D. The track efficiency

The bank efficiency is not the only component of the total track efficiency. Geometrical efficiency and fit efficiency must be considered also. Geometrical inefficiencies are generated by the segmentation of the detector and are due to boundary crossing tracks, since patterns are required to be entirely contained in the  $\Delta\phi$  sector. Quantitatively, the inefficiency is 15% at  $P_T = 2$  GeV/c, 6% at  $P_T = 5$  GeV/c, and 3% at  $P_T = 10$  GeV/c.

Fit efficiency depends on the  $\chi^2$  cut applied during the track fitting step. The fit cuts are adjusted so that the fit efficiency is always 90%. Therefore, the total efficiency for the three  $P_T$  thresholds are 69% at 2 GeV, 76% at 5 GeV and 78% at 10 GeV.

### IV. CONCLUSIONS

The data organization done by the FTK processor allows a level-2 trigger logic composed of commercial CPUs to reconstruct full resolution tracks inside roads within typical level-2 times. FTK is a very compact amount of hardware, even for very complex applications: a fourth of CMS barrel would require half 9U VME crate. It can find tracks at an event rate of 100 kHz. It is eligible for tracking data reduction in trigger applications. Hits of track candidates, with  $P_T$  above a threshold of few GeV and with impact parameters compatible with  $b$  quark decay, can be filtered among a huge

number of other hits. The ambitious goal of *trigger selection* of  $b$  decays at the future hadron colliders can benefit from our architecture.

### V. REFERENCES

- [1] A.Bardi et al., "A Real-Time Tracker for Hadronic Collider Experiments", *IEEE Trans. on Nucl.Sci.*, vol. 46, 1999 pp. 947-952.
- [2] M. Dell'Orso and L. Ristori, "VLSI structures for track finding", *Nucl. Instr. and Meth.*, vol. A278, 1989 pp. 436-440.
- [3] H. Grote, "Pattern recognition in high-energy physics", *Rep. Prog. Phys.*, vol. 50, 1987 pp. 473-500.
- [4] CMS Collaboration, "The Compact Muon Solenoid Technical Proposal", CERN/LHCC 94-38, LHCC/P1, 15 December 1994; "The Tracker Project Technical Design Report", CERN/LHCC 98-6, CMS TDR 5, March 1998.
- [5] S. Belforte et al., "SVT: An Online Silicon Vertex Tracker for the CDF Upgrade", *Nucl. Instr. and Meth.*, vol. A409, 1998 pp. 658-661.
- [6] M.G. Bagliesi et al., "The Data Organizer: a High Traffic Node for Tracking Detector Data", summary N. 514 presented to this conference.
- [7] [www.xilinx.com](http://www.xilinx.com)
- [8] M.G. Bagliesi et al., "A Pipeline of Associative Memory Boards for Track Finding", summary N. 516 presented to this conference.
- [9] A. Cisternino et al., "A Standard Cell based Content-Addressable Memory System for Pattern Recognition", Fourth Workshop on Electronics for LHC Experiments, CERN/LHCC/98-36, 30 October 1998.
- [10] International Technology Roadmap for Semiconductors, 1998 update, Semiconductor Industry Association.
- [11] The CMS Collaboration, "The Tracker Project Technical Design Report", CERN/LHCC 98-6 CMS TDR 5, march 18, 1998.
- [12] <http://www.thep.lu.se/torbjorn/pythia>.
- [13] H. Wind, "Principal Component Analysis and its Applications to Track Finding", in "Formulae and Methods in Experimental Data Evaluation", European Physical Society, vol. 3 (1984), p. K1.
- [14] R. Carosi, G. Punzi, "An algorithm for a real time track fitter", contributed paper to the 1998 IEEE Nuclear Science Symposium, 10-12 November, 1998, Toronto, Canada.
- [15] <http://www.sgi.com>